STUDENT UPI:

Question 1 (11 marks)

a) What would be printed out by the following code segment?

```
int i = 10;
double d = i / 3;
System.out.println(d);
```

3.0

(1 mark)

b) What is printed by the following?

```
System.out.println(7 + 7 + 7 + "7" + (7 + 7) + 7);
```

217147

(1 mark)

c) Write an expression to generate a random integer between 8 and 13 (including 8 and 13) and assign it to the variable `i`.

```
int i = (int)(Math.random() * 6) + 8;
```

(1 mark)

d) What is the output of the following code segment?

```
System.out.println("a");
System.out.print("a");
System.out.println("a");
System.out.print("a");
System.out.println("a" + "b");
System.out.println("a");
```

a
aa
aab
a

(2 marks)

e) What is the output of the following code segment?

```
double d = (int)(5.3 + 2.8);
System.out.println(d);
```

8.0

(1 mark)

f) What is the output of the following code segment?

```
System.out.println("\\n\"");
```

\n"

(1 mark)

g) What is the output of the following code segment?

```
int i1 = 2 + 2 - 2 * 2 / 2;
int i2 = 2 * 2 / 2 - 2 + 2;
System.out.println(i1 + "." + i2);
```

2.2

(1 mark)

h) What is the output of the following code segment:

```
int i = 9 - 8 - (7 * 6) / 5 + 4 - (3 + 2) * 2;
System.out.println(i);
```

-13

(1 mark)

i) What is the output of the following code segment:

```
String s = "computer science";
String t = s.substring(3, 6);
System.out.println(s.length() - t.length() + t);
```

13put

(2 marks)

Question 2 (8 marks)

Complete the following method which displays the question passed as a parameter and waits for the user to type a response. If the first character of the response is a "y" or a "Y" then the method should return `true`. In all other cases it should return `false`. You can assume that the `readInput()` method used in labs and assignments is available to return the `String` typed by a user.

```
public boolean askQuestion(String question) {
  System.out.print(question);
  String answer = readInput();
  String firstChar = answer.substring(0, 1);
  return firstChar.equals("Y") ||
      firstChar.equals("y");









}
```

Question 3 (5 marks)

a) Consider the following piece of Java code. After this code has been executed are the following Boolean expressions `true` or `false`?

```
int a = 37;
int b = 6;
double x = b - 1.0;
int c = (int)(a/x);
boolean myCompare = c == a / b;
```

1. `(c > b) && (a % 2 == 1)`

   true

2. `(x != 6.0) || myCompare`

   true

3. `myCompare ^ (b < (a - c)/b)`

   false

(3 marks)

b) Why is it not recommended to compare `double` values in the following manner?

```
double x = 0.2 + 0.2 + 0.2;
if (x == 0.6)
     System.out.println("We have a match!");
```

Floating point numbers, such as double, do not always represent a value exactly. Therefore exact comparisons such as with == and != can lead to unexpected false or true results.

(2 marks)

Question 4 (5 marks)

Write a method called `minimumOf4()`, which is passed four integer parameters and returns the smallest of them.

For example, once you have completed this method, the output of the following code:

```
System.out.println(minimumOf4(4, -2, 1, 7));
System.out.println(minimumOf4(2, 3, 4, 5));
System.out.println(minimumOf4(7, 17, 27, 6));
```

would be:

```
-2
2
6
```

```
public int minimumOf4 (int a, int b, int c, int d) {
  return Math.min(a, Math.min(b, Math.min(c, d)));

or
  if (a < b && a < c && a < d)
      return a;
  else if (b < c && b < d)
      return b;
  else if (c < d)
      return c;
  else return d;
}
```

Question 5 (8 marks)

Write a program that prompts the user to enter 2 integers, and then displays the product of those 2 numbers. Example output is shown below, where the user has entered the value 11 and the value 7.

```
Enter number 1:  11
Enter number 2:  7
The product is:  77
```

Your program must produce output in exactly the same format as shown above. Complete the `go()` method. You can assume that the `readInput()` method used in labs and assignments is available to return the `String` typed by a user.

```
public class Products {

  public void go() {
    System.out.print("Enter number 1: ");
    int num1 = Integer.parseInt(readInput());
    System.out.print("Enter number 2: ");
    int num2 = Integer.parseInt(readInput());
    System.out.println("The product is: " +
      (num1 * num2));
  }









}
```

Question 6 (6 marks)

What is the output of the following program, assuming the user enters the following:

```
Ian Michael Short
```

at the prompt "`What is your name?`".

```
public class Names {

  public void go() {

    System.out.print("What is your name? ");
    String name = readInput();

    int posOfSpace;

    String i1 = name.substring(0,1);
    posOfSpace = name.indexOf(" ");
    name = name.substring(posOfSpace + 1, name.length());
    String i2 = name.substring(0,1);
    posOfSpace = name.indexOf(" ");
    name = name.substring(posOfSpace + 1, name.length());

    System.out.println("Hello " + i1 + ". " + i2 + ". " + name);
  }
}
```

```
What is your name? Ian Michael Short
Hello I. M. Short
```

Question 7 (6 marks)

We wish to test whether `Math.random()` could simulate a coin toss. So we decide to call `Math.random()` 1000 times and count how many times it returns a value less than 0.5. Complete the code on the **opposite** page by adding a while loop which executes 1000 times and counts how many of the calls to `Math.random()` are less than 0.5. At the end of the while loop the variable `lessThan5` should contain the number of calls to `Math.random()` which were less than 0.5.

```
int lessThan5 = 0;
int counter = 0;
while (counter < 1000) {
  if (Math.random() < 0.5)
      lessThan5++;
  counter++;
}
```

Question 8 (8 marks)

Complete this piece of code to find the minimum and maximum values in an array of doubles using just one `while` loop. At the end of the `while` loop `minValue` should contain the minimum value in the `numbers` array and `maxValue` should contain the maximum value in the `numbers` array.

```
double minValue = numbers[0];
double maxValue = numbers[0];
int arrayIndex = 1;
while (arrayIndex < numbers.length) {
  if (numbers[arrayIndex] < minValue)
      minValue = numbers[arrayIndex];
  if (numbers[arrayIndex] > maxValue)
      maxValue = numbers[arrayIndex];
  arrayIndex++;
}
```

or

```
while (arrayIndex < numbers.length) {
  minValue = Math.min(minValue, numbers[arrayIndex]);
  maxValue = Math.max(maxValue, numbers[arrayIndex]);
  arrayIndex++;
}
```

Question 9 (10 marks)

a) Complete the following method with a `switch` statement. The method should compare `dayOfWeek` with the various day constants and set `dayType` to either `WEEKEND` or `WEEKDAY` depending upon its value. If `dayOfWeek` is not in the range from 0 to 6, it should set `dayType` to `INVALID`.

```
public int isWeekend(int dayOfWeek) {
  final int SUNDAY = 0;
  final int MONDAY = 1;
  final int TUESDAY = 2;
  final int WEDNESDAY = 3;
  final int THURSDAY = 4;
  final int FRIDAY = 5;
  final int SATURDAY = 6;

  final int WEEKEND = 0;
  final int WEEKDAY = 1;
  final int INVALID = -1;

  int dayType;

  switch (dayOfWeek) {
  case MONDAY: case TUESDAY: case WEDNESDAY:
      case THURSDAY: case FRIDAY:
      dayType = WEEKDAY;
      break;
  case SUNDAY: case SATURDAY:
      dayType = WEEKEND;
      break;
  default:
      dayType = INVALID;
  }



  return dayType;
}
```

(5 marks)

b) Rewrite the `switch` statement in part a) as a series of nested `if-else-if` statements which achieve the same result.

```
public int isWeekend(int dayOfWeek) {
  final int SUNDAY = 0;
  … same constants as above
  final int SATURDAY = 6;

  final int WEEKEND = 0;
  final int WEEKDAY = 1;
  final int INVALID = -1;
```

```
  int dayType;
  if (dayOfWeek == MONDAY || dayOfWeek == TUESDAY ||
      dayOfWeek == WEDNESDAY || dayOfWeek == THURSDAY
      || dayOfWeek == FRIDAY)
      dayType = WEEKDAY;
  else if (dayOfWeek == SUNDAY ||
      dayOfWeek == SATURDAY)
      dayType = WEEKEND;
  else
      dayType = INVALID;









  return dayType;
}
```

(5 marks)

Question 10 (5 marks)

Write JavaDoc comments for the following method. Ensure you describe each parameter as well as the return type. You should write a meaningful description of what the method accomplishes at the start of the JavaDoc.

```
/**
 * Determine whether a value is even and within
 * the two bounds specified
 *
 * @param testValue  A value to be tested
 * @param lowBound   The minimum value testValue
      could be
 * @param highBound  The maximum value testValue
      could be
 * @return           true if the value is even and
      within the bounds


 */
```

```
  public boolean boundedEvenInt(int testValue, int lowBound,
                                                int highBound) {
       return (testValue % 2 == 0) && (testValue >= lowBound) &&
                           (testValue <= highBound);
  }
```

Question 11 (5 marks)
Rewrite the following code using the standard COMPSCI101 indenting method.

```
public class Triangle {
public void printStars(int stars) { int line = 0;
while (line < stars) { int number = 0;
while (number <= line) { System.out.print("*");
number++; } System.out.println(); line++; }}}
```

```
public class Triangle {
  public void printStars(int stars) {
    int line = 0;
    while (line < stars) {
      int number = 0;
      while (number <= line) {
        System.out.print("*");
        number++;
      }
      System.out.println();
      line++;
    }
  }
}
```

Question 12 (5 marks)
Use truth tables to determine whether the following two Boolean expressions are equivalent.

1. `!(a && b)`
2. `!(a || (a && b)) && !b`

a	b	(a && b)	!(a && b)
True	True	True	False
True	False	False	True
False	True	False	True
False	False	False	True

a	b	(a&&b)	a||(a&&b)	!(a||(a&&b))	!b	!(a || (a && b)) && !b
True	True	True	True	False	False	False
True	False	False	True	False	True	False
False	True	False	False	True	False	False
False	False	False	False	True	True	True

Are the expressions equivalent?

Yes

Question 13 (10 marks)

There are 5 errors that prevent this program compiling successfully. Circle the errors and write the correction alongside.

```
/**
 * This class calculates the average of an array of numbers.
/   */
public class Average {
    /**
     * The go method is the usual start method for COMPSCI 101.
     */
    public void go() {
        int[] numbersToAverage = {1.5, 2.7, 3.1, 4.29, 5.0};
        double result = average(numbersToAverage);
        System.out.Println(result);
    }
    /**
     * Calculates the average of an array of doubles.
     *
     * @param numbers the array of doubles
     * @return the average of the numbers in the array
     */
    public double average(double[] numbers) {
        double sum = 0;
        int index = 0;
        while (index < numbers.length)
            sum = sum + numbers[index];
            index = index + 1;
        }
        sum / numbers.length;
    }
}
```

double

println

{

return

Question 14 (8 marks)

Write a method that takes three integer values as its arguments and places these values into an array which is returned from the method. The method should be named `putInArray` and can be called in the following manner:

```
int[] arrayOf3 = putInArray(-4, 1024, 12);
```

The method has to create a new array of the appropriate size; assign the input parameters to the appropriate elements in the array; and return the array. Write the complete method below.

```
public int[] putInArray(int num1, int num2, int num3)
{
  int[] tempArray = new int[3];
  tempArray[0] = num1;
  tempArray[1] = num2;
  tempArray[2] = num3;
  return tempArray;
}
```

or

```
public int[] putInArray(int num1, int num2, int num3)
{
  int[] tempArray = {num1, num2, num3};
  return tempArray;
}
```